```
pragma solidity ^0.5.3;

contract controls{


    //event

    event LogLoop(int);
    event LogBreak(string);
    //if - else

    function ifElse(uint x, uint y) public pure returns(uint result){

        if (x > y){
            result = x - y;
            return result;
        }else if (x < y){
            result = y - x;
            return result;
        }else{
            return 0;
        }
    }

    //while loop
    function whileLoop(int counter) public {
        int i = 0;
        while(i < counter){
            emit LogLoop(i);
            i = i + 1;

        }

    }

    // do - while
    function doLoop(int counter) public {
        int i = 0;
        do{
            emit LogLoop(i);
            i = i + 1;
        }while(i < counter);

    }

    //for loop

    function forLoop(int counter) public{
        for (int i = 0; i < counter; i++){
            emit LogLoop(i);
        }
    }

    //breaak

    function breakStatement(int counter) public{
        for (int i = 0; i < counter; i++){
            if (i==4)
               break;
            emit LogLoop(i);
        }
        emit LogBreak("Out of the loop");
    }

    //continue

     function continueStatement(int counter) public{
        for (int i = 0; i < counter; i++){
            if (i==3)
               continue;
            emit LogLoop(i);
        }

        int j=0;
        while (j < counter){
            j= j+1;
            if (j==3)
               continue;
            emit LogLoop(j);
        }

    }


}
```